Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 06-010

Incremental Window-based Protein Sequence Alignment Algorithms

Huzefa Rangwala and George Karypis

March 23, 2006

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
23 MAR 2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Incremental Window-based Protein Sequence Alignment Algorithms	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Computer Science and Engineering,University of Minnesota,200 Union Street SE,Minneapolis,MN,55455-0159**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**10**	

Incremental Window-based Protein Sequence Alignment Algorithms

Huzefa Rangwala and George Karypis

Department of Computer Science & Engineering, University of Minnesota,Minneapolis, MN 55455

{rangwala,karypis} @cs.umn.edu

Abstract

Motivation: *Protein sequence alignment plays a critical role in computational biology as it is an integral part in many analysis tasks designed to solve problems in comparative genomics, structure and function prediction, and homology modeling.*

Methods: *We have developed novel sequence alignment algorithms that compute the alignment between a pair of sequences based on short fixed- or variable-length high-scoring subsequences. Our algorithms build the alignments by repeatedly selecting the highest scoring pairs of subsequences and using them to construct small portions of the final alignment. We utilize PSI-BLAST generated sequence profiles and employ a profile-to-profile scoring scheme derived from PICASSO.*

Results: *We evaluated the performance of the computed alignments on two recently published benchmark datasets and compared them against the alignments computed by existing state-of-the-art dynamic programming-based profile-to-profile local and global sequence alignment algorithms. Our results show that the new algorithms achieve alignments that are comparable or better to those achieved by existing algorithms. Moreover, our results also showed that these algorithms can be used to provide better information as to which of the aligned positions are more reliable—a critical piece of information for comparative modeling applications.*

Suppl. Data *http://bioinfo.cs.umn.edu/supplements/win-aln/*

1 Introduction

Alignment algorithms serve as the most basic sequence analysis methods in computational biology and have a wide range of applications dealing with sequence database searching, comparative modeling, protein structure and function prediction.

The current state-of-the-art sequence alignment algorithms have a well defined optimal dynamic programming based solution, introduced decades ago. These optimal algorithms, Smith-Waterman [35] and Needleman-Wunsch [27] solve the local and global sequence alignment problems respectively. Over the years, alignment methods have advanced with several variations of the optimal alignment method, use of gap modeling techniques [13], heuristics [1, 29], and more recently the use of profile [12, 7, 2] and structure information [18].

In recent years, there has been a considerable research effort in developing kernel-based methods for building discriminatory models for remote homology detection and fold recognition. This research has led to the development of a number of protein *string kernels* that determine the similarity between a pair of proteins as a function of the number of sufficiently similar short subsequences that they share. These string kernels have proven to be extremely effective in building very accurate models, and these methods are among the best performing schemes for remote homology prediction and fold recognition [22, 21, 30] .

Motivated by these developments in string kernels, the work in this paper is designed to address the question as to the extent to which, ideas motivated by these string kernels can be used to build alignments between a pair of sequences. Toward this goal, we developed a set of window-based alignment algorithms that are heuristic in nature. Our methods incrementally constructed the alignment by using the highest scoring pairs of residues between the two sequences at each step. The residue pair scoring was borrowed from string kernel theory where to score the residue pairs in consideration, we examined short subsequences, referred to a $w$mers centered around each of the two residues. We introduced several heuristics to identify aligned residue pairs using the $w$mers coupled with profile information.

We determined the quality of our alignment methods by evaluation on a template-based [7, 31] and a model-based dataset [8, 5]. Our empirical results on the two datasets showed the competitive performance of our introduced schemes to state-of-the-art methods. We also evaluated our methods by determining the reliability of the aligned positions [17, 4, 32, 25, 36]. The positive results for some of our alignment algorithms on such a reliability metric is very encouraging due to far reaching applications, like comparative modeling.

2 Methods

2.1 Sequence Profiles and Profile Scoring

The alignment algorithms that we developed take advantage of evolutionary information by utilizing PSI-BLAST [2] generated sequence profiles.

The profile of a sequence X of length m is represented by two $m \times 20$ matrices. The first is its position-specific scoring matrix PSSM_X that is computed directly by PSI-BLAST using the scheme described in [2]. The rows of this matrix cor-

respond to the various positions in X and the columns correspond to the 20 distinct amino acids. The second matrix is its position-specific *frequency* matrix PSFM_X that contains the frequencies used by PSI-BLAST to derive PSSM_X. These frequencies (also referred to as *target frequencies* [26]) contain both the sequence-weighted observed frequencies (also referred to as *effective frequencies* [26]) and the BLOSUM62 [15] derived-pseudocounts [2].

Many different schemes have been developed for determining the similarity between profiles that combine information from the original sequence, position-specific scoring matrix, or position-specific target and/or effective frequencies [26, 37, 24]. In our work we use a scheme that is derived from PICASSO [14, 26] that was recently used in developing effective remote homology detection and fold recognition algorithms [30]. Specifically, the similarity score between the ith position of protein's X profile, and the jth position of protein's Y profile is given by

$$\begin{aligned} S_{X,Y}(i,j) \quad = \quad & \sum_{l=1}^{20} \text{PSFM}_X(i,l)\,\text{PSSM}_Y(j,l) \; + \\ & \sum_{l=1}^{20} \text{PSFM}_Y(j,l)\,\text{PSSM}_X(i,l), \end{aligned} \tag{1}$$

where $\text{PSFM}_X(i,l)$ and $\text{PSSM}_X(i,l)$ are the values corresponding to the lth amino acid at the ith position of X's position-specific scoring and frequency matrices. $\text{PSFM}_Y(j,l)$ and $\text{PSSM}_Y(j,l)$ are defined in a similar fashion.

2.2 Window-based Alignments

The overall methodology of the alignment algorithms developed in this work is to incrementally construct the alignment by using various heuristics to identify the pairs of aligned residues. The key idea shared by these algorithms is that they determine whether or not a pair of residues should be aligned together by examining the (short) subsequences, referred to as $w$mers, that are centered around each of the two residues.

Given a sequence X of length m and a user-supplied parameter w, the $w$mer at position i of X ($w < i \leq m - w$) is defined to be the $(2w+1)$-length subsequence of X centered at position i. That is, the $w$mer contains x_i, the w amino acids before, and the w amino acids after x_i. A pair of $w$mers are compared by computing their ungapped alignment scores. Given two sequences X and Y, the ungapped alignment score, $w\text{score}(x_i, y_j)$, between a pair of $w$mers at positions i and j of X and Y, respectively is given by

$$w\text{score}(x_i, y_j) = \sum_{k=-w}^{w} S_{X,Y}(i+k, j+k), \tag{2}$$

where $S_{X,Y}(i+k, j+k)$ is the alignment score between x_{i+k} and y_{j+k} and is computed using Equation 1.

2.2.1 Central Alignment Scheme (CA).

This is the simplest alignment algorithm that we developed and computes the alignment by progressively aligning the pairs of residues that have the highest positive $w$score values subject to the constraint that they do not conflict with the portion of the alignment that has been constructed thus far.

Specifically, given two sequences X and Y of length m and n, respectively and a value for w, it starts by computing the set $\mathcal{S}_w$ of residue-pairs that are candidates for inclusion in the alignment by considering only the pairs that have positive $w$score values. That is,

$$\mathcal{S}_w = \{(x_i, y_j) \mid w\text{score}(x_i, y_j) > 0\}, \tag{3}$$

where $w < i \leq m - w$ and $w < j \leq n - w$. Then it performs a series of iterations in which it performs the following three steps: First, it extracts from $\mathcal{S}_w$ the residue-pair with the highest $w$score value (x_{i^*}, y_{j^*}): Second, it aligns x_{i^*} against y_{j^*}: Third, it removes from $\mathcal{S}_w$ all residue-pairs that cannot be part of a valid alignment given that x_{i^*} and y_{j^*} have been aligned with each other. This process terminates when $\mathcal{S}_w$ becomes empty. Positions that do not belong to any of the selected residue pairs are left unaligned (i.e., aligned against spaces).

The residue pairs that need to be removed are: (i) (x_{i^*}, y_l) $\forall l$, (ii) (x_k, y_{j^*}) $\forall k$, (iii) (x_k, y_l) $\forall (k > i^* \wedge l < j^*)$, and (iv) (x_k, y_l) $\forall (k < i^* \wedge l > j^*)$. The first two conditions remove from $\mathcal{S}_w$ all residue-pairs involving x_{i^*} or y_{j^*}, as these positions have now been aligned, whereas the last two conditions remove the residue-pairs that if aligned, will introduce inversions in the alignment.

2.2.2 Subset Alignment Scheme (SA).

A limitation of the central alignment scheme is that it may leave a large number of residues unaligned because (i) it only considers the residue-pairs with positive $w$scores, and (ii) it will not align the first and last w positions of the two sequences ($\mathcal{S}_w$ contains only pairs involving interior residues).

To address this problem we developed the subset alignment scheme (SA), which can be considered an extension to the CA scheme. Specifically, the SA scheme modifies the second and third steps of the CA algorithm as follows. During the second step, in addition to including the (x_{i^*}, y_{j^*}) pair in the alignment, it also includes in the alignment all previously unaligned residue-pairs of the form (x_{i^*+k}, y_{j^*+k}) for $-w < k < +w$. That is, it can potentially include all residue-pairs involved in (x_{i^*}, y_{j^*})'s $w$mer. Note that due to the incremental nature of the algorithm, the second step essentially extends the alignment around the (x_{i^*+k}, y_{j^*+k}) residue-pair until it encounters a residue (from either X or Y) that has already been aligned. We will refer to this as the *alignment extension* operation. During the third step the SA algorithm removes from $\mathcal{S}_w$ all residue-pairs that are now in conflict with all aligned residue-pairs that were selected in second step.

2.2.3 Central and Subset Alignment Scheme (CSA).

A potential problem with the SA scheme, is that it may align a pair of residues (x_{i^*+k}, y_{j^*+k}) with each other, even when $\mathcal{S}_w$ contains residue-pairs with higher $w$score values for either or both of the two residues. This happens, because SA's alignment extension operation extends the alignment as soon as it extracts the highest scoring residue pair from $\mathcal{S}_w$ and there may be some higher-scoring $w$mers for these positions in $\mathcal{S}_w$.

For this reason, we developed a hybrid scheme that combines the CA and SA approaches. Specifically, the new scheme first computes a CA alignment and then augments it by applying the alignment extension approach used by SA to each pair of its aligned residues.

2.2.4 Variable $w$mer Alignment Scheme.

The alignment schemes, CA, SA, and CSA were discussed in the context of a fixed length $w$mer. The potential drawback of this scheme is that if w is set to a relatively large value, it may fail to identify positive scoring subsequences; whereas if it is set too low, it may fail to reward residue-pairs that have relatively long similar subsequences.

For this reason we extended the algorithms to also operate with variable length $w$mers. The key difference from the use of fixed length $w$mers centered around residue pairs x_i and y_j is the fact that we define length w^* in the range of 1 to w, such that

$$w^* = \underset{\mathcal{K}=1}{\overset{w}{\operatorname{argmax}}}\ \mathcal{K}\text{score}(x_i, y_j), \tag{4}$$

where $\mathcal{K}$score is the $(2\mathcal{K}+1)-$subsequence score as defined in Equation 2.

Our alignment schemes start by computing the set $\mathcal{S'}_w$ of residue pairs that are candidates for inclusion in the alignment by considering only pairs that have positive w^*score values. With this change all steps of our alignment algorithms remain same. Note that the SA scheme using the variable length $w$mers will have its alignment extension operation extended till a maximum length of w^*.

As a notation reference we denote the variable $w$mer alignment algorithms by CA^v, SA^v, and CSA^v to distinguish them from the fixed $w$mer alignment algorithms denoted in this study by CA^f, SA^f, and CSA^f.

3 Materials

3.1 Evaluation Methodologies and Metrics

We evaluated the performance of the proposed window-based alignment algorithms by considering (i) the quality of the alignment itself and (ii) the extent to which the inherent ordering of the aligned pairs of residues can be used to identify portions of the alignment that are more reliable than others. In order to assess alignment quality we used two widely used methodologies, often referred to as template-based [7] and model-based [8], whereas the reliability was assessed by following a methodology that was recently proposed in the context of comparative modeling [36].

3.1.1 Template-based Approach.

The first method for evaluating alignment quality compares the differences between the alignment generated to template alignments [7, 31, 8]. These template alignments are generally derived from various structural alignment programs and are considered to be the gold standard.

We use three quality scores, namely the developer's score (f_D) [31], the modeler's score (f_M) [31] and the Cline score (CS) [4] to compare the template alignments with the generated alignments. The developer's score is the number of correctly aligned residue pairs in the generated alignment divided by the length of the template alignment. (The *length* of an alignment is defined as the number of aligned residue pairs.) The modeler's score computes the ratio of correctly aligned residue pairs with the length of the generated alignment. The Cline score was developed to address the issues with f_M and f_D by penalizing both under-alignment and over-alignment, and also crediting regions in the generated alignment that may be shifted by a few positions relative to the reference alignment [7, 4]. The steps for computation of the Cline score can be found in the study [4].

Note that the f_D and f_M scores are equivalent to the more traditional measures of *recall* and *precision* [9], respectively that are used extensively to measure prediction performance. In the rest of the discussion we will primarily refer to f_D and f_M by the more intuitive names of recall and precision, respectively.

3.1.2 Model-based Approach.

An alternative to using a template-based approach is to build a structural model from the alignment and evaluate the similarity between the model and the template structure [8, 28]. Starting from the alignment between a pair of proteins (one protein considered to be the query protein, the second considered to be the target protein whose 3D structure is known), a model protein is created which consists of the carbon alpha, C_α atoms of the query protein. The atomic coordinates of this model protein are the atomic coordinates of the target protein i.e., for every aligned pair of residues, the query protein has its C_α atomic coordinates replaced by the corresponding atomic coordinates of the target protein. The similarity between the two structures (the model protein and target protein) after a structural super-imposition [23], is used as an assessment of sequence alignment quality.

In our study, we computed this similarity using the LGscore [5] that takes into account the common segments between the pair of proteins. LGscore computes the similarity between two protein structures (model and template structure) based on the common segments between them. It is desirable to have long common segments with high structural similarity. The LGscore measure was used to evaluate the structures obtained by threading methods [28] in the CAFASP2 [10] and LiveBench [3] experiments as well as a sequence alignment quality measure [8].

Note that instead of LGscore other structural similarity methods or protein modeling assessment measures can be

used for evaluating the quality of the model (e.g rmsd measure [19], global distance test score (GDT) [38] and MaxSub [34]). However, for this study we show only the results using the LGscore method due to similarity in results obtained when tested with the other measures.

3.1.3 Reliability of Aligned Regions. In comparative modeling and several other applications, it is essential not only to align residue pairs but also to provide some reliability index or confidence measure associated with the aligned residue pairs. While building protein structure models using comparative modeling strategies it is important to include only those regions where the alignment is considered to be good or reliable [17, 4, 32, 25, 36].

One of the reliability assessment measures calculated a smoothed profile-derived alignment score. The score for each of the aligned residue in the template alignment was computed using a triangular smoothing window of size 5. The reliability was assessed by setting up a threshold value for the smoothed profile-derived score [36]. Our approach for reliability assessment was very similar to this method.

Using the template-based benchmarks we evaluated the reliability of the aligned residue pairs by ranking the aligned pairs in the query alignment. We score the aligned positions using fixed length $w$scores. The reliability measure is computed as the recall at different percent levels of incorrectly aligned residue pairs (up to 5%). The notion of a hit is defined as having the same aligned residue pairs in both the query and template alignments. The difference in our reliability scheme was the use of a profile-to-profile scoring functions equally weighted at all positions of the $w$mer rather than using a smoothing $w$mer [36].

3.2 Datasets

For the template-based assessment scheme we used a dataset created to evaluate the various profile-to-profile scoring functions for protein sequence alignment [7]. The dataset consists of 588 reference alignment pairs having high structural similarity but low sequence identity ($\leq$ 30%). This dataset was selected to have a high pairwise structural similarity using the consensus of FSSP [16] and CE [33].

For the model-based evaluation scheme, we used a benchmark created from SCOP 1.39 filtered to only contain domains with less than 50% pairwise sequence identity [8]. This dataset contains of 9983 protein domain pairs, such that 1903 belong to the same families, 3101 share only the same superfamily, and 4979 share only the same fold. Due to the non-symmetrical nature of models built from alignments, each pair of sequences were evaluated twice—leading to a benchmark of 19966 domain pairs.

3.3 Profile Generation

The position specific score and frequency matrices used by the profile-based scoring method of Equation 1 were generated using the latest version of the PSI-BLAST algorithm (available in NCBI's blast release 2.2.10), and were derived from the multiple sequence alignment constructed after five iterations using an e value of 10^{-3}. The PSI-BLAST was performed against NCBI's nr database that was downloaded in November of 2004 and contained 2,171,938 sequences.

In the case in which PSI-BLAST could not produce meaningful alignments for certain positions of the query sequence, the corresponding rows of the two matrices are derived from the scores and frequencies of BLOSUM62.

4 Results

In this section, we evaluate the performance of the incremental window based alignment schemes using the various benchmark datasets and evaluation metrics discusses in Section 3.

4.1 Assessment of Incremental Window-based Alignments

Table 1 provides an extensive set of results illustrating the performance of the CA, SA, and CSA schemes on the template-based dataset for different values of w and for fixed- and variable-length $w$mers. Note that the column labeled "$CS_{\leq 15\%}$" shows the CS results for the subset of sequence-pairs that have less than 15% sequence identity (i.e., a subset that is inherently harder to align well).

4.1.1 Central vs Subset vs Combined. The results of Table 1 show that with respect to the CS scores, SA tends to perform better than either CA or CSA, whereas CA performs consistently the worst. The only exception is for variable-length $w$mers, in which SA's performance is comparable to that of CSA. The relative advantage of SA is more evident if we consider the subset of sequence-pairs with less that 15% sequence identity, for which its CS scores are consistently higher than those achieved by the other schemes (SA achieves a score of 0.649 whereas CA and CSA achieves scores of 0.614 and 0.628, respectively).

By looking at the performance of the various schemes in terms of recall, we can see that SA's higher CS-based performance is due to the fact that it achieves significantly better recall values than the other schemes. This was to be expected, as it was one of the motivation behind the development of SA. Also, the precision-based results show that CA achieves somewhat better precisions than CSA, whereas SA's precision is comparable or better to that of the other schemes.

4.1.2 Fixed vs Variable Length Alignments. Analyzing the performance of alignment methods that use fixed length $w$mers compared to the methods that use variable length $w$mers, we notice that for the CA and CSA schemes, for the same $w$mer length the recall as well as the precision scores have higher values. Note that the higher recall is expected, because the methods using a variable $w$mer size window will have a higher flexibility in allowing larger number of $w$mers (with a positive score) to be picked for the candidate set $\mathcal{S'}_w$.

Another key observation is that SA^f performs better in

terms of recall than SA^v. This is because for the same value of w, the w^* value selected by SA^v may be smaller than w (i.e., the value used by SA^f). As a result, SA^f's alignment extension operations will involve longer windows, which can produce longer alignments than SA^v, and thus higher recall values.

Table 1: Alignment Accuracy Results on a Template-based Dataset.

	f_M (precision)	f_D (recall)	CS	$CS_{\leq 15\%}$
fixed				
central				
$w$mer = 2	0.805	0.791	0.803	0.600
$w$mer = 3	0.799	0.776	0.794	0.596
$w$mer = 4	0.791	0.756	0.782	0.587
$w$mer = 5	0.776	0.732	0.764	0.572
subset				
$w$mer = 2	0.802	0.835	0.826	0.626
$w$mer = 3	0.805	0.842	0.831	0.642
$w$mer = 4	0.805	0.842	0.832	0.644
$w$mer = 5	0.802	0.838	0.828	0.649
combined				
$w$mer = 2	0.791	0.822	0.816	0.619
$w$mer = 3	0.785	0.819	0.814	0.623
$w$mer = 4	0.779	0.811	0.808	0.624
$w$mer = 5	0.767	0.798	0.798	0.624
variable				
central				
$w$mer = 2	0.799	0.804	0.809	0.595
$w$mer = 3	0.802	0.807	0.812	0.605
$w$mer = 4	0.805	0.797	0.810	0.611
$w$mer = 5	0.805	0.797	0.807	0.614
subset				
$w$mer = 2	0.798	0.827	0.820	0.615
$w$mer = 3	0.798	0.834	0.825	0.629
$w$mer = 4	0.798	0.836	0.827	0.634
$w$mer = 5	0.794	0.832	0.823	0.636
combined				
$w$mer = 2	0.795	0.822	0.813	0.600
$w$mer = 3	0.797	0.827	0.820	0.614
$w$mer = 4	0.800	0.831	0.824	0.621
$w$mer = 5	0.800	0.832	0.825	0.628

In the table f_M denotes the Modeler's score, f_D denotes the Developer's score, CS denotes the Cline score, and $CS_{\leq 15\%}$ denotes the Cline score for a subset of sequence pairs sharing less than 15% sequence identity.

4.1.3 Sensitivity of Schemes with respect to varying $w$mer size Looking at the performance achieved by the various schemes in Table 1 as w ranges from two to five, we see that in general, SA's and CSA's performance does not significantly change (e.g., CS scores stay within a tight range), whereas CA^f's performance tend to deteriorate with increasing w. This latter behavior is due to the fact that as we increase the $w$mer size, fewer $w$mers will have a positive score and hence will not be included as part of the set $\mathcal{S}_w$. We see a direct effect of this leading to a decrease in the recall scores. Also increase in the $w$mer size does lead to a decrease in precision score as well. This is because for a larger $w$mer window the positive scoring $w$mers may not be due to the more "central" positions. Evidence of this can be seen by comparing the behavior of the CA^v scheme in which both the precision and recall scores stay the same.

Another key observation is that the schemes that utilize variable length $w$mers tend to perform better for larger values of w. This is because of the flexibility associated with using a variable length $w$mer.

4.1.4 Alternative Performance Assessment For this dataset too, we performed a thorough parameter study by varying $w$mer lengths for our alignment schemes. We observed similar results as seen in T:B1 for the template-based dataset. In Table 2 we report only the best results achieved rather than showing results for varying $w$mer sizes as done in Table 1.

Firstly, we notice the difference in the LGscore values for the family, superfamily and fold pairs clearly showing the difficulty nature of the three sets of problems, with the fold-pairs being the hardest to model followed by the superfamily and family level pairs.

Similar to the template-based results, the SA scheme has the best LGscore at the family, superfamily and fold levels for both the variable and fixed $w$mer setting. A surprising fact was that the performance results as measured by the LGscore did not decrease with increasing $w$mer lengths. In fact, we observed that the use of a higher $w$mer size of 5 for the fixed length scheme achieved the best results of 1.53 and 4.29 for the fold and superfamily level problems. We also observe slightly better performance for the variable $w$mer schemes compared to the fixed $w$mer schemes.

The performance of the CSA^v alignment method was the lowest for both the family and superfamily level pairs which contrasts the results seen previously on the template-based dataset in Table 1.

4.2 Comparison with Earlier Results

4.2.1 Template-based Benchmark. Table 3 shows the comparative performance of our window based schemes against some of the best profile-to-profile scoring techniques studied previously [7]. In the table we show results for the schemes pdotp, correlp and coach. pdotp uses dot product to compute the similarity between two profiles, correlp computes the Pearson correlation between the profile columns, whereas coach [6] uses an asymmetrical complex dot product between the HMM profile and a position frequency matrix.

We show results of these schemes as published previously [7] using SAM T99 profiles (The performance of these

Table 2: Alignment Accuracy Results on a Model-based Dataset.

Alignment Scheme	Family	Superfamily	Fold
CA^f (2)	14.86	1.66	0.04
SA^f (5)	16.44	4.29	1.53
CSA^f (2)	15.47	2.53	0.203
CA^v (5)	15.10	2.43	0.12
SA^v (5)	16.48	4.05	1.05
CSA^v (5)	14.05	2.32	0.14

The numbers in the parameter indicate the $w$mer length for the various alignment schemes.

alignment methods using SAM T99 profiles is 3-4% better than the PSI-BLAST based profiles [7]) Our methods show comparable performance to these alignment methods using SAM T99 templates.

We also compare the results of the window based alignment methods to a local Smith-Waterman [35] alignment algorithm implementation (SW-PSSM) using the same profile-to-profile scoring function as used for the window based alignments (Equation 1). Within this local alignment framework we use an affine gap model along with a zero-shift parameter [37] to maintain certain necessary requirements of a good optimal alignment. We optimize the gap modeling parameters (gap opening (*go*), gap extension (*ge*)) and the zero shift value (*zs*) to obtain highly optimal alignments for comparative purposes.

We observe in Table 3 that the incremental window-based alignment schemes perform very competitively when compared to our fully optimized SW-PSSM implementation. Also notice the superiority of our optimized SW-PSSM implementation to the alignment methods using pdotp, correlp and coach as their profile-profile scoring functions. The difference in the SW-PSSM results with the other standard alignment techniques may be due to the use of a more sensitive PICASSO based profile-to-profile scoring function. Further, these results verify that we are comparing our novel window based alignment methods to a fully optimized SW-PSSM alignment algorithm.

The performance of the window-based scheme is actually very promising. We select one of the better performing schemes (SA^f) and compare it to the optimized SW-PSSM algorithm using the CS score. Figure 1 shows that the comparative performance of the two methods across the 588 alignment pairs in the dataset.

4.2.2 Model-based Benchmark. Our results in Table 4 reiterate the closeness in performance of the incremental window based alignment method to the highly optimized SW-PSSM alignment algorithm for the family, superfamily and fold level subsets.

Table 4 also shows results for the optimized local (local sequence alignment using a global scoring matrix), global

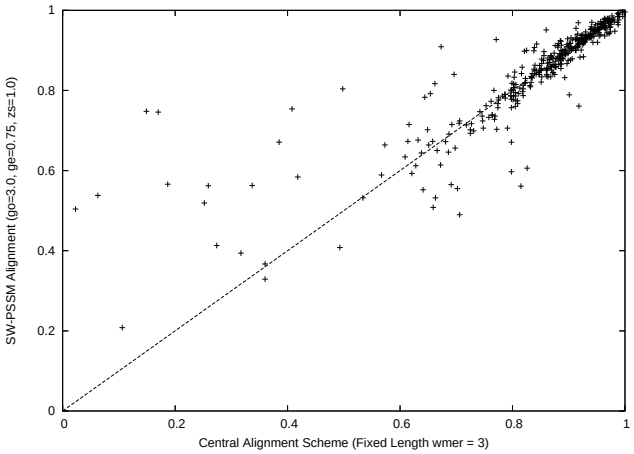

Figure 1: Cline Score Comparison of SW-PSSM scheme against SA^f scheme for the 588 alignment pairs in the template-based dataset

Table 3: Comparative Performance with Earlier Results on Template-based Dataset.

Alignment Scheme	f_M	f_D	CS	$CS_{\leq 15\%}$
SA^f (3)	0.805	0.842	0.831	0.642
SA^v (4)	0.798	0.836	0.827	0.634
SW-PSSM	0.803	0.852	0.841	0.689
pdotp (T99)	0.806	0.829	0.832	0.697
correlp (T99)	0.794	0.835	0.829	0.702
coach (T99)	0.797	0.830	0.829	0.697

The optimized SW-PSSM results are achieved using $go = 3.0$, $ge = 0.75$, $zs = 1.0$. In the table pdotp, correlp, coach use a dot product, correlation function, and a HMM based profile-profile scoring function. T99 denotes the use of SAM T99 based profiles respectively.

(global sequence alignment using a global scoring matrix), PSI (3D-PSSM [20] based global sequence alignment against a profile [11] obtained from PSI-BLAST), SSPSI [8](3D-PSSM based global sequence alignment against a profile obtained from PSI-BLAST using secondary structure information) and structural (alignment using structural superimposition by lgscore2) alignment methods published previously [8]. The structural alignment sets up a higher reference quality score for the benchmark. Using sequence alignment techniques we would like to achieve these high levels of accuracy. The results shown in Table 4 for the various previously published schemes, as well as for our methods are the best achieved after optimization of the various parameters.

We further analyze the data by annotating a model as being correct based on the LGscore value. As done in the study [8] we use the less strict LGscore cutoff (10^{-3}) to define a correct model and a more stringent cutoff (10^{-5}) to identify models of higher quality. The percentage of models correct based

on these cutoffs are shown in Table 5. Both the incremental window-based alignment methods, as well as the SW-PSSM alignment method, are able to pick the correct models with similar degrees of accuracy. Our techniques also seem to identify a higher percentage of correct models when compared to the previously studied schemes, especially PSI and SSPSI, both of which also incorporate some profile information. As seen from Table 5 our methods are able to pick a larger fraction of higher quality models for the family and superfamily levels.

4.2.3 Reliability Performance. Table 6 shows the reliability performance for the window based alignment schemes in comparison to the optimized SW-PSSM based alignment scheme. These results correspond to the average recall scores obtained for all the alignment pairs at different error rates using the procedure described in Section 3.1.3.

Though the SW-PSSM algorithm showed slightly better performance in terms of the overall alignment quality (Table 3 and Table 4), it is interesting to note the window-based schemes using variable length $w$mers showed far better performance at the lower error rates. In particular before seeing any incorrect predictions in the ranked aligned positions, the alignment methods using variable length $w$mers have a recall around 0.260 compared to the recall of 0.205 for the SW-PSSM algorithm. Note that the recall performance of the CSA scheme is slightly better than the CA scheme and slightly worse compared to the SA alignment scheme. These results can be explained by the fact that the high scoring residue pairs aligned by CA are also aligned by the CSA scheme.

Table 4: Comparative Performance with Earlier Results on a Model-based Dataset.

Alignment Scheme	Family	Superfamily	Fold
SA^f (5)	16.44	4.29	1.53
SA^v (5)	16.48	4.05	1.05
SW-PSSM	16.66	4.38	2.02
local	14.1	2.0	0.7
global	15.1	2.9	1.4
PSI	15.8	3.3	1.4
SSPSI	16.0	4.1	2.6
structural	19.4	9.1	8.0

The optimized SW-PSSM results are achieved using $go = 3.0$, $ge = 0.75$, $zs = 3.0$. All the results are optimized for their relevant parameters

5 Conclusion

In this study we developed algorithms that identify the aligned pairs of residues using an incremental approach. These algorithms capture the most similar pairs of subsequences as part of the final alignment. The concepts from string-kernel theory (use of ungapped subsequences, scored using profiles) played an integral role in the design of these alignment algorithms.

Table 5: Fraction of Correct Models based on the LGscore.

LGscore	$< 10^{-3}$			$< 10^{-5}$		
Alignment Scheme	Fm	Sf	Fd	Fm	Sf	Fd
SA^f (3)	74	27	5	55	8	0
SA^v (3)	74	28	4	55	8	0
SW-PSSM	74	27	6	56	8	0
local	66	10	1	46	2	0
global	70	12	1	49	3	0
PSI	72	18	4	50	4	0
SSPSI	73	21	6	53	5	0
structural	86	60	51	66	21	21

The optimized SW-PSSM results are achieved using $go = 3.0$, $ge = 0.75$, $zs = 3.0$. All the results are optimized for their relevant parameters. Fm, Sf and Fd denote the family-level, superfamily-level and fold-level performance results respectively.

Table 6: Reliability Assessment: Recall for the first $k\%$ errors.

Method	0%	1%	2%	3%	4%	5%
CA^f (3)	0.176	0.281	0.365	0.434	0.494	0.541
SA^f (3)	0.186	0.297	0.384	0.459	0.519	0.563
CSA^f (3)	0.180	0.286	0.370	0.438	0.498	0.545
CA^v (3)	0.254	0.364	0.450	0.515	0.566	0.603
SA^v (3)	0.260	0.368	0.454	0.521	0.572	0.612
CSA^v (3)	0.260	0.367	0.454	0.520	0.571	0.610
SW-PSSM	0.205	0.320	0.405	0.480	0.541	0.586

The optimized SW-PSSM results are achieved using $go = 3.0$, $ge = 0.75$, $zs = 3.0$. The numbers in the parenthesis represent the $w$mer width used for the results shown.

Our comprehensive experimental study on the template-based and model-based benchmark datasets showed comparable performance to a fully optimized Smith-Waterman profile-based implementation. In terms of the reliability performance of the aligned residue-pairs we notice that the alignment schemes using variable length $w$mers had very promising results. Amongst the window-based schemes we noticed that the subset alignment, SA using both the fixed and variable $w$mers showed the best performance. The sensitivity analysis done by varying the $w$mer size showed the SA schemes to have a robust performance.

The simplicity of our methods and competitive alignment quality as well as aligned region reliability will lead to the application of our algorithms in key bioinformatic problems,

especially comparative modeling.

6 Acknowledgment

We would like to express our deepest thanks to Professor Arne Elofsson and Dr. Robert C. Edgar for providing us their datasets and codes for the study. This work was supported by NSF EIA-9986042, ACI-0133464, IIS-0431135, NIH RLM008713A, the Army High Performance Computing Research Center contract number DAAD19-01-2-0014, and by the Digital Technology Center at the University of Minnesota.

References

[1] S. F. Altschul, W. Gish, E. W. Miller, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] S. F. Altschul, L. T. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–402, 1997.

[3] J. Bujnicki, A. Elofsson, D. Fischer, and L. Rychlewski. Livebench: continuous benchmarking of protein structure prediction servers. *Protein Science*, 10:352–361, 2001.

[4] M. Cline, R. Hughey, and K. Karplus. Predicting reliable regions in protein sequence alignments. *Bioinformatics*, 18:306–314, 2002.

[5] S. Cristobal, A. Zemla, D. Fischer, L. Rychlewski, and A. Elofsson. A study of quality measured for protein threading models. *BMC Bioinformatics*, 2(5), 2001.

[6] R. Edgar and K. Sjolander. Coach: profile-profile alignment of protein families using hidden markov models. *BIOINFORMATICS*, 20(8):1309–1318, 2004.

[7] R. Edgar and K. Sjolander. A comparison of scoring functions for protein sequence profile alignment. *BIOINFORMATICS*, 20(8):1301–1308, 2004.

[8] A. Elofsson. A study on protein sequence alignment quality. *PROTEINS:Structure, Function and Genetics*, 46:330–339, 2002.

[9] T. Fawcett. Roc graphs: Notes and practical considerations for researchers, 2004.

[10] D. Fischer, A. Elofsson, L. Rychlewski, F. Pazos, A. Valencia, B. Rost, A. Ortiz, and R. L. Dunbrack. Cafasp2: The second critical assessment of fully automated structure prediction methods. *PROTEINS: Structure, Function and Genetics*, 45(5):171–183, 2001.

[11] M. Gribskov, A. D. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *PNAS*, 84:4355–4358, 1987.

[12] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computational Chemistry*, 20:25–33, 1996.

[13] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, 1997.

[14] A. Heger and L. Holm. Picasso:generating a covering set of protein family profiles. *Bioinformatics*, 17(3):272–279, 2001.

[15] S. Henikoff and J. G. Henikoff. Amino acid subsitution matrices from protein blocks. *PNAS*, 89:10915–10919, 1992.

[16] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273(5275):595–602, 1996.

[17] L. Jaroszewski, W. Li, and A. Godzik. In search for more accurate alignments in the twilight zone. *Protein Science*, 11:1702–1713, 2002.

[18] D. T. Jones, W. R. Taylor, and J. M. Thorton. A new approach to proetin fold recognition. *Nature*, 358:86–89, 1992.

[19] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32(5):922–923, 1996.

[20] L. Kelley, R. MacCallum, and M. Sternberg. Enhanced genome annotation using structural profiles in the program 3d-pssm. *Journal of Molecular Biology*, 299:523–544, 2000.

[21] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Computational Systems Bioinformatics*, pages 152–160, 2004.

[22] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[23] M. Levitt and M. Gerstein. A unified statistical framework for sequence comparison and structure comparison. *Proceedings of National Academy of Science, USA*, 95:5913–5920, 1998.

[24] M. Marti-Renom, M. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Science*, 13:1071–1087, 2004.

[25] H. T. Mevissen and M. Vingron. Quantifying the local reliability of sequence alignment. *Protein Engineering*, 9(2):127–132, 1996.

[26] D. Mittelman, R. Sadreyev, and N. Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19(12):1531–1539, 2003.

[27] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[28] A. Panchenko, A. Marchler-Bauer, and S. H. Bryant. Combination of threading potentials and sequence profiles improves fold recognition. *Journal of Molecular Biology*, 296:1319–1331, 2000.

[29] William R. Pearson and David J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85:2444–2448, 1988.

[30] H. Rangwala and G. Karypis. Profile based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247, 2005.

[31] J. M. Sauder, J. W. Arthur, and R. L. Dunbrack. Large-scale comparison of protein sequenece alignments with structural alignments. *Proteins*, 40:6–22, 2000.

[32] M. Schlosshauer and M. Ohlsson. A novel approach to local reliability of sequence alignments. *Bioinformatics*, 18(6):847–854, 2002.

[33] I. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Engineering*, 11:739–747, 1998.

[34] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer. Maxsub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785, 2000.

[35] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[36] M. L. Tress, D. Jones, and A. Valencia. Predicting reliable regions in protein alignments from sequence profiles. *Journal of Molecular Biology*, 330:705–718, 2003.

[37] G. Wang and R. L. Dunbrack JR. Scoring profile-to-profile sequence alignments. *Protein Science*, 13:1612–1626, 2004.

[38] A. Zemla. Lga: A method for finding 3d similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374, 2003.